CLAIMS

What is claimed is:

1. A method for providing communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol, the method comprising the steps of:

a) establishing a communication channel between the first network node and the second network node;

b) establishing a first stream between the first process and the communication channel;

c) establishing a second stream between the second process and the communication channel;

d) encrypting data to be transmitted between the first and second processes, the encrypting of the data being independent of the at least one communication protocol supported by the first network node;

e) writing the encrypted data to the first stream;

f) causing the encrypted data to be transmitted from the first network node to the second network node;

g) reading the encrypted data from the second stream; and

h) decrypting the encrypted data to obtain decrypted data which is identical to the data on the first network node before the data was encrypted.

2. The method of Claim 1, further including the steps of

a) performing a communication protocol-specific encryption of the data on the first network node, and

b) performing a communication protocol-specific decryption of the data on the second network node.

3. The method of Claim 1, wherein the communication channel is a Java secure channel,

wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and

wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

4. The method of Claim 1, wherein the communication channel is a Java secure channel, wherein the first stream is a Java stream, wherein the second stream is a Java stream, wherein the method further comprises the step of connecting the Java secure channel to a third Java stream, and wherein the third Java stream provides for the transmission of data according to a specific communication protocol.

5. A computer-readable medium having stored thereon a plurality of sequences of instructions for providing communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol, the plurality of sequences of instructions including sequences of instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

a) establishing a communication channel between the first network node and the second network node;

b) establishing a first stream between the first process and the communication channel;

c) establishing a second stream between the second process and the communication channel;

d) encrypting data to be transmitted between the first and second processes, the encrypting of the data being independent of the communication protocols supported by the first network node;

e) writing the encrypted data to the first stream;

f) causing the encrypted data to be transmitted from the first network node to the second network node;

g) reading the encrypted data from the second stream; and

h) decrypting the encrypted data to obtain decrypted data which is identical to the data on the first network node before the data was encrypted.

6. The computer-readable medium of Claim 5, wherein the computer-readable medium further includes instructions for performing the steps of

a) performing a communication protocol-specific encryption of the data on the first network node, and

b) performing a communication protocol-specific decryption of the data on the second network node.

7. The computer-readable medium of Claim 5, wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and

wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

8. The computer-readable medium of Claim 5, wherein the communication channel is a Java secure channel,

wherein the first stream is a Java stream,

wherein the second stream is a Java stream,

wherein the computer-readable medium further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol.

9. A communication network providing communication protocol-independent secure communication between a first network node and a second network node, wherein the first network node and the second network node each support at least one common communication protocol, wherein the first network node is communicatively coupled to the second network node by a communication channel, the communication network comprising:

a) a first process executing on the first network node, wherein the first process provides for the communication protocol-independent encryption of data;

b) a first stream which provides for the transfer of encrypted data between the first process and the communication channel;

c) a second process executing on the second network node; and

d) a second stream which provides for the transfer of encrypted data between the communication channel and the second process, wherein the second process also provides for the decryption of data which has been encrypted by the first process.

10. The communication network of Claim 9, wherein the second process further includes the capability to decrypt data based upon any communication protocol supported by the second network node.

11. The communication network of Claim 9, wherein the communication channel is a Java secure channel, the first stream is a Java stream and the second stream is a Java stream.

12. The communication network of Claim 11, further comprising a third Java stream connected to the Java secure channel, the third Java stream providing for the transmission of data according to a specific communication protocol.

13. A computer data signal embodied in a carrier wave and representing sequences of instruction which, when executed by one or more processors, provide communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol by performing the steps of:

a) establishing a communication channel between the first network node and the second network node;

b) establishing a first stream between the first process and the communication channel;

c) establishing a second stream between the second process and the communication channel;

d) encrypting data to be transmitted between the first and second processes, the encrypting of the data being independent of the communication protocols supported by the first network node;

e) writing the encrypted data to the first stream;

f) causing the encrypted data to be transmitted from the first network node to the second network node;

g) reading the encrypted data from the second stream; and

h) decrypting the encrypted data to obtain decrypted data which is identical to the data on the first network node before the data was encrypted.

14. The computer data signal of Claim 13, wherein the computer sequence of instructions further includes instructions for performing the steps of

a) performing a communication protocol-specific encryption of the data on the first network node, and

b) performing a communication protocol-specific decryption of the data on the second network node.

15. The computer data signal of Claim 13, wherein the first stream is a first Java stream, wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and

wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

16. The computer data signal of Claim 13, wherein the communication channel is a Java secure channel,

wherein the first stream is a Java stream,

wherein the second stream is a Java stream,

wherein the computer sequence of instructions further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol.

17. A method for providing communication protocol-independent security for data transmitted by a process executing on a network node, the method comprising the steps of:

a) establishing a stream between the process and a communication channel;

b) encrypting data to be transmitted by the process, the encrypting of the data being independent of a communication protocol supported by the network node;

c) writing the encrypted data to the stream; and

d) causing the encrypted data to be transmitted from the network node to the communication channel.

18. The method of Claim 17, wherein the communication channel is a Java secure channel,
wherein the stream is a first Java stream,
wherein the step of establishing a stream between the process and the communication channel further comprises the step of establishing a Java stream between the process and the Java secure channel, and
wherein the step of writing the encrypted data to the stream further comprises the step of writing the encrypted data to the Java stream.

19. The method of Claim 17, wherein the communication channel is a Java secure channel, wherein the stream is a Java stream,
wherein the method further comprises the step of connecting the Java secure channel to a second Java stream, and
wherein the second Java stream provides for the transmission of data according to a specific communication protocol.